

REMARKS/ARGUMENTS

The amendments to the specification correct clearly typographical errors. No new matter has been added by any of the amendments to the specification.

Claims 1-23 are pending in the present application. Claims 12 and 13 are cancelled. Claims 2-5, 7-8, 11, 14, 16, 17, 19-21, and 23 are amended. Claims 24 and 25 are added. Support for the new claims and claim amendments can be found in the claims themselves and in the Applicant's patent application on page 9, lines 9-17, page 12, lines 6-15, page 13, line 11 - page 15, line 12, and in Figures 4 and 5.

I. 35 U.S.C. § 101

The Examiner rejected claims 8 and 17-23 under 35 U.S.C. § 101 as being directed towards non-statutory subject matter. Regarding claim 17, Applicants have amended the claim accordingly, thereby overcoming the rejection. Because claims 18-23 depend from claim 17, the rejection has also been overcome with respect to claims 18-23.

Regarding claim 8, the Examiner states:

As per claim 8, the wording of claim appears to be directed to a non-functional descriptive material, data *per se*, stores (for example on a medium). What's being stored on the (storage or medium) does not appear to meet the IEEE definition of a data structure, and no instructions or other code for causing functionality that results in a practical application appears to be present. As such, claim 8 appears to be solely non-functional descriptive material stored on a medium and therefore non-statutory.

Office Action dated December 14 2006, p. 2.

Amended claim 8 is as follows:

8. A file system, in a recordable-type computer readable medium, comprising:
a data store, wherein the data store stores associations between files and unique identifiers; and
a file management process, wherein the file management process associates the unique identifier with the file in the data store when a file is saved and identifies files associated with a unique identifier in the data store when a query to retrieve files using the unique identifier is made.

Amended claim 8 is statutory because it claims functional descriptive material that is recorded on a recordable-type computer readable medium. When functional descriptive material is recorded on a computer-readable medium, such as a recordable-type medium, the functional descriptive material becomes structurally and functionally interrelated to the medium and will be statutory in most cases. Use of technology permits the function of the descriptive material to be realized. MPEP 2106.01. In this context, "functional descriptive material" consists of data structures and computer programs which impart

functionality when employed as a computer component. Id. The MPEP defines a “data structure” as “a physical or logical relationship among data elements, designed to support specific data manipulation functions.” Id. Furthermore, a claimed computer-readable medium encoded with a data structure defines structural and functional interrelationships between the data structure and the computer software and hardware components which permit the data structure’s functionality to be realized, and is thus statutory. Id. In contrast, “nonfunctional descriptive material” includes but is not limited to music, literary works, and a compilation or mere arrangement of data. Id.

Amended claim 8 claims functional descriptive material because claim 8 claims a data structure, as defined by the MPEP. First, claim 8 claims a logical relationship among the data elements because the file management process associates a unique identifier with a file in a data store. The relationship between unique identifiers and files in a data store facilitates identifying files in a data store, and is therefore logical. Second, the relationship between unique identifiers and files in a data store is designed to support specific data manipulation functions because claim 8 uses the relationship between unique identifiers and files to identify files associated with a unique identifier in the data store when a query to retrieve files using the unique identifier is made. Thus, amended claim 8 defines a data structure.

Further, the data structure defined in amended claim 8 imparts functionality when employed as a computer component. Specifically, claim 8 provides the functionality of identifying files associated with a unique identifier in the data store when a query to retrieve files using the unique identifier is made. Because amended claim 8 imparts functionality when employed as a computer component, amended claim 8 claims functional descriptive material.

In addition, the functional descriptive material claimed in amended claim 8 is recorded on a recordable-type computer readable medium. Therefore, because amended claim 8 claims functional descriptive material recorded on a recordable-type computer-readable medium, amended claim 8 claims statutory subject matter. Accordingly, the rejection of claim 8 has been overcome.

II. 35 U.S.C. § 103, Obviousness; Claim 1-4, 7-13, 16-20, and 23

The Examiner rejected claims 1-4, 7-13, 16-20, and 23 under 35 U.S.C. § 103 as obvious over *Howard*, Multiple Display File Directory and File Navigation System for a Personal Computer, U.S. Patent 6,185,574, November 26, 1997 (hereinafter “*Howard*”) in view of *Peltonen*, Method and System for Effectively Representing Query Results in a Limited Amount of Memory, U.S. Patent 5,926,807, May 8, 1997 (hereinafter “*Peltonen*”). Claims 12 and 13 have been cancelled. Therefore, the rejection of claims 12 and 13 is now moot. The rejection as applied to claims 1-4, 7-11, 16-20, and 23 is respectfully traversed.

With regard to claims 1, 8, 10, and 17, the Examiner states that:

As per claims 1, 8, 10, and 17, Howard et al. disclose a system, which including data processing system for locating files in a hierarchical directories as detailed in col. 10, lines 10-12, col. 12, lines 27-34, col. 12, line 66 through col. 14, line 6). In particular, Howard et al. disclose the claimed limitations wherein an input has been received indicating that a file is to be saved (See Howard et al. Fig. 15 in conjunction with steps 754 and 758, col.31, lines 18-23). Howard et al. specifically disclose that a user can retrieve a file where the file is saved as detailed in col.4, lines 55-57, col. 12, lines 16-27). It is noted, however, Howard et al. did not specifically disclose the system for saving the file in association with a unique identifier in a data store, responsive to receiving an input, wherein the data store describes associations between files and unique identifiers and wherein files are retrieved based on unique identifiers. On the other hand, Peltonen et al. disclose a system for effectively representing query result in a memory where files have been saved or stored (See Peltonen et al. Title and abstract, col.2, lines 55-67). In particular, Peltonen et al. achieved the claimed limitations of, saving the file in association with a unique identifier in a data store, responsive to receiving an input, wherein the data store describes associations between files and unique identifiers and wherein files are retrieved based on unique identifiers, by providing a bookmark associated with files as a unique identifier to identify files (See Peltonen et al. Abstract lines 14-19, col.1.9, lines 25-67, col. 11, lines 16-39).

It would have been obvious to one of ordinary skill in the art at the time of applicant's invention was made to modify the parallel virtual directory system of Howard et al. by incorporating the bookmark mechanism taught by Peltonen et al. because that would have enhanced the system of Howard et al. by allowing it to retrieve files or documents store in data store quickly and efficiently using the bookmark as a unique identifier and return the result appropriately in response to a request (See Peltonen et al. Abstract lines 14-19, col.1.8, lines 1-6).

Office Action dated December 14, 2006, pp. 4-5.

Claim 1, from which claims 2-4 and 7 depend, and which is representative of claims 8-10 and 17, is as follows:

1. A method in a data processing system for locating files, the method comprising:
 - receiving an input indicating that a file is to be saved; and
 - responsive to receiving the input, saving the file in association with a unique identifier in a data store, wherein the data store describes associations between files and unique identifiers and wherein files are retrieved based on unique identifiers.

No *prima facie* obvious rejection has been made against claim 1 because neither *Howard* nor *Peltonen* teach or suggest all of the features of claim 1. Further, no proper motivation, suggestion, or teaching exists to combine *Howard* and *Peltonen* to achieve the invention of claim 1.

II.A. Neither *Howard* nor *Peltonen* Teach All of the Features of Claim 1

No *prima facie* obviousness rejection can be stated because neither reference, alone or in combination, teaches all of the features of claim 1. The Examiner bears the burden of establishing a *prima facie* case of obviousness based on the prior art when rejecting claims under 35 U.S.C. § 103. *In re Fritch*, 972 F.2d 1260, 23 U.S.P.Q.2d 1780 (Fed. Cir. 1992). Additionally, all limitations of the claimed invention must be considered when determining patentability. *In re Lowry*, 32 F.3d 1579, 1582, 32 U.S.P.Q.2d 1031, 1034 (Fed. Cir. 1994). Therefore, the Examiner fails to state a *prima facie* obviousness rejection if the proposed combination does not teach all of the features of the claimed invention.

A *prima facie* obviousness rejection has not been stated because the proposed combination of the references does not teach all of the features of claim 1. Specifically, neither *Howard* nor *Peltonen* teach the feature of responsive to receiving the input, saving the file in association with a unique identifier in a data store. The Examiner admits that *Howard* does not teach this claimed feature. Additionally, because *Howard* is devoid of disclosure in this regard, nothing in *Howard* suggests this claimed feature.

However, the Examiner asserts that *Peltonen* teaches this claimed feature, citing various portions of *Peltonen*. This assertion is mistaken because *Peltonen* fails to teach the feature of receiving an input indicating that a file is to be saved, and therefore cannot teach the “responsive to” feature in the claimed feature. Each of the cited portions will be addressed in turn to show that *Peltonen* does not teach or suggest this claimed feature. Applicants first address the following portion of *Peltonen*:

In a further embodiment of the invention, when a request is received to return the row data for a row among the identified subset of rows, the retained bookmarks for the identified subset of rows are used to retrieve row data for the rows back into the result set. The requested row data is then returned from the result set in response to the request.

Peltonen, Abstract, lines 14-19.

Neither the cited portion nor any other portion of *Peltonen* teaches the feature of responsive to receiving the input, saving the file in association with a unique identifier in a data store. *Peltonen* discloses a method and system for internally representing database query results using a limited amount of memory. The cited portion discloses the retrieval of row data, which is returned from a result set in response to a request for row data. However, the cited portion does not relate to any saving function, let alone teach or suggest an input indicating that a file is to be saved, as required by the claimed feature.

On the other hand, claim 1 recites the feature of responsive to receiving the input, saving the file in association with a unique identifier in a data store. Claim 1 further defines an input as one that indicates that a file is to be saved. Claim 1 links the input indicating that a file is to be saved with saving the file in association with a unique identifier in a data store. In other words, the feature of saving is in response to receiving the input. The request in the cited portion of *Peltonen* is not the same as the input,

as claimed, because the request in *Peltonen* is a request to return data to an identified subset of rows for which data was previously removed due to memory constraints, while the input, as claimed, indicates that a file is to be saved. The request does not indicate that any data is to be saved because the data is already stored someplace and now needs to be retrieved. For example, the cited portion discloses a request that is “received to return the row data for a row among the identified subset of rows.” (*Peltonen*, Abstract, lines 15-16) However, the cited statement discloses only a request to return row data that is already stored someplace, and fails to teach or suggest that a file is to be saved, as claimed.

Peltonen’s failure to teach or suggest an input that a file is to be saved is expected because *Peltonen*’s invention focuses on the logical representation of database query results using a limited amount of memory and the retrieval of those results, and not on the interface with an outside source as it relates to saving files. Further, because *Peltonen* fails to teach or suggest the feature of receiving the input indicating that a file is to be saved, *Peltonen* cannot teach or suggest that anything is “responsive to” receiving the input, let alone the feature of saving the file in association with a unique identifier in a data store. Therefore, *Peltonen* fails to teach or suggest the claimed feature of responsive to receiving the input, saving the file in association with a unique identifier in a data store.

Nevertheless, the Examiner also cites the following portion of *Peltonen*:

The table object preferably further contains a bookmark hash table 313, which maps a bookmark to the segment object containing the row having that bookmark. The concept of hash tables and various effective implementations thereof are well known to those skilled in the art. In an alternative embodiment, to reduce memory requirements, bookmark hash table 313 is omitted from the table object, and the table object queries the segment objects in turn to identify the segment containing a row having a particular bookmark. To facilitate this process, in accordance with the alternative embodiment, each segment supports a ContainsBookmark method that, when invoked on the segment, indicates whether the segment contains a row having the specified bookmark. The table object further contains a cursor position bookmark 314 that identifies a position within the sorted result set, and is used to return rows sequentially in key value order to the row sink program.

FIG. 4 is a segment object diagram showing the contents of segment object 340 in greater detail. Segment object 440 is a full segment object and therefore contains complete information for each of its rows, including a bookmark sort vector 491 and key sort vector 492 used to identify the different orders for traversing the row data. The row data is arranged in a row store 480. In one preferred embodiment, variable-length information for the rows is stored in a different area of the segment than fixed-length information for the rows (not shown). Row data for a newly added row is appended to the end of the row store. Each row in the row store is divided into a number of columns: a bookmark column 481 contains the bookmark for a row, which may be used by the row source program to quickly re-retrieve row data for the row; a key column 482 contains the primary key for sorting rows in the result set; and additional data columns 483 and 484 each contain further information relating to each row. The bookmark sort vector contains a pointer to each of the rows in the row store. The

pointers are arranged in the same order as bookmarks for the rows, so that, when the bookmark sort vector is traversed from top to bottom, the bookmark of each row pointed to is larger than the bookmark of the previously visited row. The key sort vector performs the same function with respect to key values instead of bookmarks.

Peltonen, column 9, lines 25-67.

However, the above cited portion fails to teach or suggest the feature of responsive to receiving the input, saving the file in association with a unique identifier in a data store. The cited portion discloses a query that is used to identify rows having a particular bookmark. The second paragraph of the cited portion describes the contents of a segment object. However, the cited portion fails to teach or suggest an input indicating that a file is to be saved, as claimed. For example, the cited portion states that the “table object queries the segment objects in turn to identify the segment containing a row having a particular bookmark.” (*Peltonen*, column 9, lines 32-34) However, a query to identify the segment object having a particular bookmark is not the same as an input indicating that a file is to be saved, as claimed, because the query relates to searching rows of data for a particular bookmark and does not relate to saving data. The remainder of the cited portion describes the contents of a segment object, and does not relate to inputs at all, let alone to inputs indicating that a file is to be saved as claimed. Once again, because *Peltonen* fails to teach or suggest the feature of receiving the input indicating that a file is to be saved, *Peltonen* cannot teach or suggest that anything is “responsive to” receiving the input, let alone the feature of saving the file in association with a unique identifier in a data store as claimed.

Lastly, the Examiner cites the following portion of *Peltonen*:

FIG. 10 is a segment object diagram showing the partial segment object that has been converted from a full segment object in greater detail. FIG. 10 shows that the partial segment object 1040 contains only the following: a first key value 1041, a last key value 1042, a row counter 1043, a sorted flag 1045, and a bookmark column 1081. The bookmark column 1081 contains a list of the bookmarks of each of the rows represented by the segment object. This list is sorted in increasing key value. The value of the sorted flag 1045 (“TRUE”) indicates that this sorted order of the bookmarks has been so far maintained. However, if additional rows are later added to the partial segment object, their bookmarks will merely be appended at the end of the list, as, without the key values for the rows already in the partial segment object, it is impossible to determine where in the key order the new row falls. The sorted flag will therefore be set to FALSE when further new rows are added to the partial segment object if the new row's key value is less than the last key value. If row data is later requested for rows in the partial segment object 1040, the partial segment object will be converted back to a full segment object by, for each bookmark in the bookmark list 1081, requesting the complete row data for the row having that bookmark from the row source program.

Peltonen, column 11, lines 16-39.

However, the cited portion fails to teach or suggest the feature of responsive to receiving the input, saving the file in association with a unique identifier in a data store. Instead, the cited portion describes a partial segment object that has been converted from a full segment object. *Peltonen* uses partial segment objects to conserve memory because they contain less data than a full segment object. If row data is requested for a partial segment object, the partial segment object can be converted back to a full segment object. The cited portion of *Peltonen* fails to teach or suggest any input indicating that a file is to be saved. For example, the cited portion discloses that “[i]f row data is later requested for rows in the partial segment object 1040, the partial segment object will be converted back to a full segment object by, for each bookmark in the bookmark list 1081, requesting the complete row data for the row having that bookmark from the row source program.” (*Peltonen*, column 11, lines 34-39 (emphasis added)) However, the requests in the cited statement are not the same as an input, as claimed, because the disclosed requests are requests for partial segment objects and row data for the partial segment object, which differs from an input indicating that a file is to be saved. The requests do not indicate that any data is to be saved because the partial segment objects and row data are already stored someplace and now need to be retrieved. Once again, because *Peltonen* fails to teach or suggest the feature of receiving the input indicating that a file is to be saved, *Peltonen* cannot teach or suggest that anything is “responsive to” receiving the input, let alone the feature of saving the file in association with a unique identifier in a data store. Therefore, *Peltonen* fails to teach or suggest the feature of responsive to receiving the input, saving the file in association with a unique identifier in a data store.

As admitted by the Examiner, “*Howard et al.* did not specifically disclose the system for saving the file in association with a unique identifier in a data store, responsive to receiving an input.” (Office Action dated December 14, 2006, p. 4) Furthermore, given the absence of disclosure in *Howard* in this regard, nothing in *Howard* suggests this claimed feature. As shown above, *Peltonen* does not teach or suggest all of the features of claim 1. Therefore, the proposed combination of *Peltonen* and *Howard*, when considered as a whole, does not teach all of the features of claim 1. For this reason, the Examiner has failed to state a *prima facie* obviousness rejection against claim 1 and the corresponding dependent claims.

II.B. The Examiner Failed to State a Proper Teaching, Motivation, or Suggestion to Combine *Howard* and *Peltonen*

In addition, no *prima facie* obviousness rejection can be made against claim 1 because no proper teaching or suggestion to combine the references has been stated. A *prima facie* case of obviousness is established when the teachings of the prior art itself suggest the claimed subject matter to a person of ordinary skill in the art. *In re Bell*, 991 F.2d 781, 783, 26 U.S.P.Q.2d 1529, 1531 (Fed. Cir. 1993). A

proper *prima facie* case of obviousness cannot be established by combining the teachings of the prior art absent some teaching, incentive, or suggestion supporting the combination. *In re Napier*, 55 F.3d 610, 613, 34 U.S.P.Q.2d 1782, 1784 (Fed. Cir. 1995); *In re Bond*, 910 F.2d 831, 834, 15 U.S.P.Q.2d 1566, 1568 (Fed. Cir. 1990). No such teaching or suggestion is present in the cited references and the Examiner has not pointed out any proper teaching or suggestion that is based on the prior art.

In the case at hand, the Examiner has stated only a proposed advantage to combining the references. Specifically, the Examiner states that:

It would have been obvious to one of ordinary skill in the art at the time of applicant's invention was made to modify the parallel virtual directory system of Howard et al. by incorporating the bookmark mechanism taught by Peltonen et al. because that would have enhanced the system of Howard et al. by allowing it to retrieve files or documents store in data store quickly and efficiently using the bookmark as a unique identifier and return the result appropriately in response to a request (See Peltonen et al. Abstract lines 14-19, col.8, lines 1-6).

Office Action dated December 14, 2006, p. 5.

However, an advantage is not necessarily a teaching, suggestion, or motivation. Further, the references themselves do not suggest the proposed advantage. In the present case, neither *Howard* nor *Peltonen* teach or suggest incorporating bookmarks into *Howard's* parallel virtual directory system. Nonetheless, the Examiner cites various portions of *Peltonen* to support the combination of references. Each of these portions will be addressed in turn. Applicants first address the following portion of *Peltonen*:

In a further embodiment of the invention, when a request is received to return the row data for a row among the identified subset of rows, the retained bookmarks for the identified subset of rows are used to retrieve row data for the rows back into the result set. The requested row data is then returned from the result set in response to the request.

Peltonen, Abstract, lines 14-19.

However, the cited portion discloses only using bookmarks to retrieve data in a database, and fails to teach or suggest combining the disclosed bookmarks with a parallel virtual directory system, such as that disclosed by *Howard*. Second, the Examiner cites the following portion of *Peltonen*:

In response to the request from the table object, the full segment object provides the requested row data. When the complete row data for a requested row is contained in one of the segment objects, this segment object satisfies row data requests from the table object by merely returning this complete row data.

Peltonen, column 8, lines 1-6.

However, the cited portion discloses only returning data from a segment object in response to a request from the table object. The cited portion fails to teach or suggest bookmarks at all, let alone combining bookmarks with a parallel virtual directory system, such as that disclosed by *Howard*.

Accordingly, the Examiner has not actually stated a teaching, motivation, or suggestion based on the references to combine the references. Furthermore, the Examiner has not provided any suggestion to combine these disparate references based on other known art to achieve the inventions of the claims. Instead, the Examiner has only put forth a hypothetical advantage of combining the references based on the Examiner's opinion, rather than on a pre-existing teaching, motivation, or suggestion found in the references themselves or found in other known art at the time of the invention. Thus, the Examiner has failed to state a *prima facie* obviousness rejection against claim 1.

II.C. Howard and Peltonen Each Solve Different Problems

In addition, one of ordinary skill would not combine the references to achieve the invention of claim 1 because the references are directed towards solving different problems. It is necessary to consider the reality of the circumstances--in other words, common sense--in deciding in which fields a person of ordinary skill would reasonably be expected to look for a solution to the problem facing the inventor. *In re Oetiker*, 977 F.2d 1443 (Fed. Cir. 1992); *In re Wood*, 599 F.2d 1032, 1036, 202 U.S.P.Q. 171, 174 (CCPA 1979). In the case at hand, the cited references address distinct problems. Thus, no common sense reason exists to establish that one of ordinary skill would reasonably be expected to look for a solution to the problem facing the inventor. Accordingly, no teaching, suggestion, or motivation exists to combine the references and the Examiner has failed to state a *prima facie* obviousness rejection of claim 1.

Howard is directed toward the problems of alleviating the rigid structure of conventional directory structures, eliminating the arbitrary organization barrier presented by the coupling of physical storage devices, and fixating the transient nature of hierarchical directories. For example, *Howard* provides that:

Several key problems still exist with conventional directory structures. The first of these is the somewhat rigid nature of the directory structure once such a structure is defined; which, for example requires that unique file names be used as the identifier within a directory. A second problem is the somewhat arbitrary organizational barrier that is presented by the coupling of physical storage devices and their related sizes to the hierarchical directory structures commonly used in computer storage; which for example, limits the number of files stored in a directory by the storage size of the physical storage size rather than permitting the user to include additional files whose file data would exceed the physical storage limitations of the physical storage device. A third problem is the transient nature of the hierarchical directory structure encountered when using removable or remote storage; which, for example, is often the result of the directory information for a storage medium only being present when the storage medium is

connected to the system. These problems make it difficult for both computer users and data management programs to organize and efficiently work with a computer file system, if at all.

Howard, column 3, lines 20-40.

On the other hand, *Peltonen* is directed to the problem of effectively representing query results given a limited amount of memory. For example, *Peltonen* provides that:

Given the drawbacks of these conventional approaches, a scheme for effectively representing query results in a limited amount of memory would be desirable.

Peltonen, column 1, lines 50-52.

Based on the plain disclosures of the references themselves, the references address completely distinct problems that are unrelated to each other. The problems of alleviating the rigid structure of conventional directory structures, eliminating the arbitrary organization barrier presented by the coupling of physical storage devices, and fixating the transient nature of hierarchal directories are completely distinct from the problem of effectively representing query results given a limited amount of memory.

Because the references address completely distinct problems, one of ordinary skill would have no reason to combine or otherwise modify the references to achieve the invention of claim 1. Thus, no proper teaching, suggestion, or motivation exists to combine the references in the manner suggested by the Examiner.

II.D. Conclusion as to Obviousness

Because claim 1 is representative of claims 8-10 and 17, the same distinctions between claim 1 and the references apply to claims 8-10 and 17. Further, the references may not be combined in the manner proposed by the Examiner in order to achieve the inventions of claims 8-10 and 17. Accordingly, no *prima facie* obviousness rejection can be made against claims 1, 8-10, and 17. Therefore, the Examiner cannot state a *prima facie* obviousness rejection against claims 2-4, 7, 11, 16, 18-20, and 23, at least by virtue of their dependency on claims 1, 8-10, and 17. Consequently, Applicants have overcome the obviousness rejection of claims 1-4, 7-11, 16-20, and 23 under 35 U.S.C. § 103.

III. 35 U.S.C. § 103, Obviousness; Claim 5, 14, and 21

The Examiner rejected claims 5, 14, and 21 under 35 U.S.C. § 103 as obvious over *Howard* in view of *Peltonen*. This rejection is respectfully traversed.

No *prima facie* obviousness rejection can be made against claims 5, 14, and 21 because neither *Howard* nor *Peltonen* teach or suggest all of the claimed features of claims of the corresponding independent claims. Further, *Howard* and *Peltonen* may not be properly combined to achieve the inventions of claims 5, 14, and 21 for the reasons provided below.

III.A. Neither *Howard* nor *Peltonen* Teach All of the Features of Claims 5, 14, and 21

No *prima facie* obviousness rejection can be made against claims 5, 14, and 21 because neither *Howard* nor *Peltonen* teach or suggest all of the features of claims 5, 14, and 21. Amended claim 5, which is representative of amended claims 14 and 21, is as follows:

5. The method of claim 1, wherein the input is a user input to save the file.

With regard to claim 5, the Examiner states that:

As per claims 5, 14, and 21, most of the limitations of these claims have been noted in the rejection of claims 1, 10, and 17. Applicant attention is directed to the rejection of claims 1, 10, and 17 above. In addition, the combination of *Howard et al.* and *Peltonen et al.*, as modified, discloses the claimed limitations, wherein input is a user input to save the file (See *Howard et al.* col.31, lines 18-23).

Office Action dated December 14, 2006, pp. 6-7.

Because claim 5 depends from claim 1, the same distinctions between the references and the claimed invention in claim 1 can also be made for claim 5. Additionally, claim 5 recites other additional combinations of features not taught by the references.

Neither *Howard* nor *Peltonen* teach or suggest the feature wherein the input is a user input to save the file. The Examiner cites the following portion of *Howard*:

receiving a save file command directed to the virtual file system for a file to be represented on the virtual directory;
initiating a save file command directed to a storage medium where the file is to be stored; and
storing the file on the storage medium.

Howard, column 31, 18-23.

Neither the cited portion nor any other portion of *Howard* teach or suggest the feature of claim 5 wherein the input is a user input to save the file. The cited portion of *Howard* discloses save file commands that are directed to a virtual file system and a storage medium. However, the cited portion fails to disclose that the save file commands are user input, as claimed.

On the other hand, claim 5 recites the feature wherein the input is a user input to save the file. The save file commands in the cited portion of *Howard* differ from the claimed feature because *Howard* does not disclose the source of the save file commands, let alone that the save file commands are user input, as claimed. Instead, *Howard* discloses only the “receiving” and “initiating” of save file commands without disclosing their source. Because *Howard* fails to teach or suggest that the save file commands are user input, *Howard* fails to teach or suggest the feature wherein the input is a user input to save the file.

Peltonen does not cure *Howard*'s lack of disclosure. *Peltonen* focuses on the logical representation of database query results using a limited amount of memory and the retrieval of those results, and not on the interface with an outside source as it relates to saving files. *Peltonen* fails to teach

or suggest the feature wherein the input is a user input to save the file. Therefore, the proposed combination of *Howard* and *Peltonen*, when considered as a whole, does not teach or suggest all of the features of claim 5. Accordingly, the Examiner has failed to state a *prima facie* obviousness rejection against claims 5, 14, and 21.

III.B. *Howard* and *Peltonen* Cannot be Properly Combined to Achieve the Inventions of Claims 5, 14, and 21

Because the Examiner attempts to combine *Howard* and *Peltonen* to reject claims 5, 14, and 21 as obvious, and because claims 5, 14, and 21 depend from claims 1, 10, and 17, the same analysis applied in Section II regarding the improper combination of *Howard* and *Peltonen* also applies to claims 5, 14, and 21. Accordingly, no *prima facie* obviousness rejection can be made against claims 5, 14, and 21.

IV. 35 U.S.C. § 103, Obviousness; Claim 6, 15, and 22

The Examiner rejected claims 6, 15, and 22 under 35 U.S.C. § 103 as obvious over *Howard* in view of *Peltonen*. This rejection is respectfully traversed.

No *prima facie* obviousness rejection can be made against claims 6, 15, and 22 because neither *Howard* nor *Peltonen* teach or suggest all of the claimed features of claims 6, 15, and 22. Further, *Howard* and *Peltonen* may not be properly combined to achieve the inventions of claims 6, 15, and 22, as described above with respect to claim 1 and for the reasons provided below.

IV.A. Neither *Howard* nor *Peltonen* Teach All of the Features of Claims 6, 15, and 22

No *prima facie* obviousness rejection can be made against claims 6, 15, and 22 because neither *Howard* nor *Peltonen* teach or suggest all of the features of claims 6, 15, and 22. Claim 6, which is representative of claims 15 and 22, is as follows:

6. The method of claim 1, wherein the input is from a program initiating saving of the file.

With regard to claim 6, the Examiner states that:

As per claims 6, 15, and 22, most of the limitations of these claims have been noted in the rejection of claims 1, 10, and 17. Applicant attention is directed to the rejection of claims 1, 10, and 17 above. In addition, the combination of *Howard et al.* and *Peltonen et al.*, as modified, discloses the claimed limitations, wherein the input is from a program initiating saving of the file (See *Howard et al.* col.3, lines 64-66, col.31, lines 21-22).

Office Action dated December 14, 2006, p. 7.

Because claim 6 depends from claim 1, the same distinctions between the references and the claimed invention in claim 1 can also be made for claim 6. Additionally, claim 6 recites other additional combinations of features not taught by the references.

Neither *Howard* nor *Peltonen* teach or suggest the feature wherein the input is from a program initiating saving of the file. The Examiner asserts otherwise, citing various portions of *Howard*. Each of these portions will be addressed in turn. Applicants first address the following portion of *Howard*:

initiating a save file command directed to a storage medium where the file is to be stored; and

Howard, column 31, lines 21-22.

Neither the cited portion nor any other portion of *Howard* teach or suggest the feature of claim 6, wherein the input is from a program initiating saving of the file. The cited portion of *Howard* discloses a save file command that is directed to a storage medium. However, the cited portion fails to teach or suggest the source of the initiated save file command.

On the other hand, claim 5 recites the feature wherein the input is from a program initiating saving of the file. The save file command in the cited portion of *Howard* differs from the claimed feature because *Howard* does not teach or suggest the source of the save file command, let alone that the save file command is from a program initiating saving of the file, as claimed. Instead, *Howard* discloses only the “initiating” of a save file command without disclosing the initiating source. Because *Howard* fails to teach or suggest that the save file command is from a program initiating saving of a file, *Howard* fails to teach or suggest the feature wherein the input is from a program initiating saving of the file, as claimed.

Nevertheless, the Examiner also cites the following portion of *Howard*:

In addition, application programs may continue to use the file storage system without modification.

Howard, column 3, lines 64-66.

However, the cited portion fails to teach or suggest any input relating to the saving of a file, let alone input from a program initiating saving of a file. The cited portion discloses only that application programs may continue to use a file storage system, but fails to disclose any saving function. Therefore, *Howard* fails to teach or suggest the feature wherein the input is from a program initiating saving of the file.

Peltonen does not cure *Howard*’s lack of disclosure. *Peltonen* focuses on the logical representation of database query results using a limited amount of memory and the retrieval of those results, and not on the interface with an outside source as it relates to saving files. *Peltonen* fails to teach or suggest the feature wherein the input is from a program initiating saving of the file. Therefore, the proposed combination of *Howard* and *Peltonen*, when considered as a whole, does not teach or suggest all of the features of claim 6. Accordingly, the examiner has failed to state a *prima facie* obviousness rejection against claims 6, 15, and 22.

IV.B. *Howard* and *Peltonen* Cannot be Properly Combined to Achieve the Inventions of Claims 6, 15, and 22

Because the Examiner attempts to combine *Howard* and *Peltonen* to reject claims 6, 15, and 22 as obvious, and because claims 6, 15, and 22 depend from claims 1, 10, and 17, the same analysis applied in Section II regarding the improper combination of *Howard* and *Peltonen* also applies to claims 6, 15, and 22. Accordingly, no *prima facie* obviousness rejection has been stated against claims 6, 15, and 22.

V. New Claims 24 and 25

New claims 24 and 25 have been added. No *prima facie* obviousness rejection can be made against claims 24 and 25 because neither *Howard* nor *Peltonen* teach or suggest all of the claimed features of claims 24 and 25. Further, *Howard* and *Peltonen* may not be properly combined to achieve the inventions of claims 24 and 25, as described above with respect to claim 1 and for the reasons provided below.

V.A. Neither *Howard* nor *Peltonen* Teach All of the Features of Claims 24 and 25

No *prima facie* obviousness rejection can be made against claims 24 and 25 because neither *Howard* nor *Peltonen* teach or suggest all of the features of claims 24 and 25. Claim 24, from which claim 25 depends, is as follows:

24. The method of claim 1, wherein the data store contains a first file that is associated with a first item, wherein the data store contains a second file that is associated with a second item, wherein the first item is the same as the second item, and wherein the first file and the second file may be retrieved based on the first item.

Because claim 24 depends from claim 1, the same distinctions between the references and the claimed invention in claim 1 can also be made for claim 24. Additionally, claim 24 claims other additional combinations of features not taught by the references.

Neither *Howard* nor *Peltonen* teach or suggest the feature wherein the data store contains a first file that is associated with a first item, wherein the data store contains a second file that is associated with a second item, wherein the first item is the same as the second item, and wherein the first file and the second file may be retrieved based on the first item. *Peltonen* discloses bookmarks that are associated with rows of data to facilitate locating and retrieving data from rows that have a particular bookmark. However, *Peltonen* fails to teach or suggest two or more rows of data that each have the same bookmark. For example, *Peltonen* discloses the following figure to illustrate bookmarks:

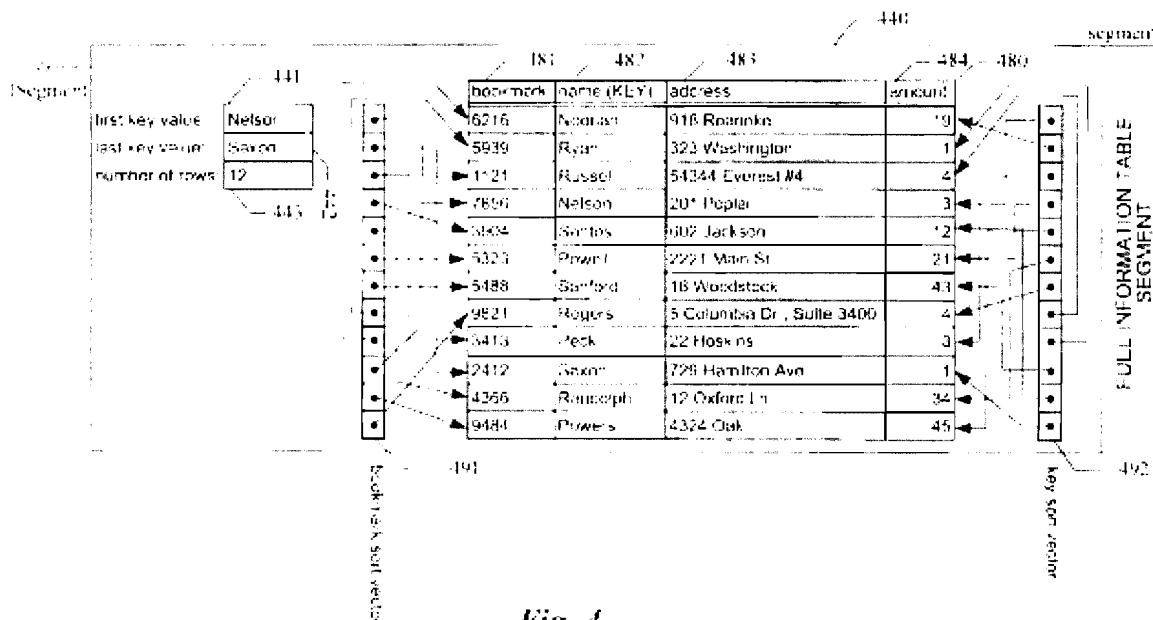


Fig. 4

Peltonen, Figure 4.

The cited figure does not teach or suggest two or more rows of data that each have the same bookmark. Indeed, *Peltonen*'s failure to teach or suggest two or more rows that have the same bookmark is expected because doing so would defeat the entire purpose of *Peltonen*'s bookmarks, which is to specify the location of a particular row for retrieval.

On the other hand, claim 24 recites the feature wherein the data store contains a first file that is associated with a first item, wherein the data store contains a second file that is associated with a second item, wherein the first item is the same as the second item, and wherein the first file and the second file may be retrieved based on the first item. The disclosure of *Peltonen* differs from the claimed feature because *Peltonen* fails to disclose two or more rows having the same bookmark, while the claimed feature recites two files that have the same item, such that the two files may be retrieved based on the item. Therefore, *Peltonen* fails to teach or suggest the feature wherein the data store contains a first file that is associated with a first item, wherein the data store contains a second file that is associated with a second item, wherein the first item is the same as the second item, and wherein the first file and the second file may be retrieved based on the first item.

Howard does not cure *Peltonen*'s lack of disclosure. *Howard* discloses a parallel virtual directory system for organizing files on a computer system, but fails to teach or suggest the feature wherein the data store contains a first file that is associated with a first item, wherein the data store contains a second file that is associated with a second item, wherein the first item is the same as the second item, and wherein the first file and the second file may be retrieved based on the first item. Therefore, the proposed

combination of *Peltonen* and *Howard*, when considered as a whole, does not teach or suggest all of the features of claim 24. Also, the Examiner cannot state a *prima facie* obviousness rejection against claim 25, at least by virtue of its dependency on claim 24.

V.B. *Howard* and *Peltonen* Cannot be Properly Combined to Achieve the Inventions of Claims 24 and 25

Because claims 24 and 25 depend from claim 1, the same analysis applied in Section II regarding the improper combination of *Howard* and *Peltonen* also applies to claims 24 and 25. Accordingly, no *prima facie* obviousness rejection can be made against claims 24 and 25.

VI. Conclusion

The subject application is patentable over the cited references and should now be in condition for allowance. The Examiner is invited to call the undersigned at the below-listed telephone number if in the opinion of the Examiner such a telephone conference would expedite or aid the prosecution and examination of this application.

DATE: March 14, 2007

Respectfully submitted,

/Theodore D. Fay III/

Theodore D. Fay III
Reg. No. 48,504
Yee & Associates, P.C.
P.O. Box 802333
Dallas, TX 75380
(972) 385-8777
Attorney for Applicants

TF/KA